Avoiding Interruptions - QoE Trade-offs in Block-coded Streaming Media Applications

Ali ParandehGheibi, Muriel Médard, Srinivas Shakkottai, Asu Ozdaglar
parandeh@mit.edu, medard@mit.edu, sshakkot@tamu.edu, asuman@mit.edu

***Abstract*—We take an analytical approach to study Quality of user Experience (QoE) for video streaming applications. First, we show that random linear network coding applied to blocks of video frames can significantly simplify the packet requests at the network layer and save resources by avoiding duplicate packet reception. Network coding allows us to model the receiver's buffer as a queue with Poisson arrivals and deterministic departures. We consider the probability of interruption in video playback as well as the number of initially buffered packets (initial waiting time) as the QoE metrics. We characterize the optimal trade-off between these metrics by providing upper and lower bounds on the minimum initial buffer size, required to achieve certain level of interruption probability for different regimes of the system parameters. Our bounds are asymptotically tight as the file size goes to infinity.**

I. Introduction

Peer-to-peer networks (P2P) are a fast-growing means of video delivery. It has been estimated that between 35-90% of Internet bandwidth is consumed by P2P applications [1], [2]. Today, P2P file-sharing networks are seeing a drop in popularity [3], but the original file sharing ideas are being used for video streaming in networks such as PPLive [4] and QQLive [5]. As smart phones become the medium of choice for Internet media access, P2P video distribution over the wireless medium is likely to gain significance.

P2P video streaming is accomplished by dividing the video file into *blocks,* which are then further divided into packets for transmission. After each block is received, it can be played out by the receiver. In order to ensure smooth sequential playout, a fresh block must be received before the current block has been played. If such a fresh block is not available the frame freezes, causing a negative user experience. Blocks may be buffered in advance of playing out in order to provide a level of protection against a frame freeze, with more initial buffering providing a lower likelihood of frame freeze.

There are two main approaches to P2P video streaming, namely, (i) using push-based multicast trees, and (ii) using pull-based mesh P2P networks. Push-based multicast trees require that each entering user should join one or more multicast trees [6], [7], [8]. Each block is pushed along a multicast tree to ensure that each user obtains blocks sequentially and with an acceptable delay. However, such an approach often involves excessive infrastructural overheads, and peer churn causes inefficiencies [9]. Pull-based mesh P2P has recently seen significant usage as a means of video delivery. Here peers maintain a playout buffer and pull blocks from each other. The approach is similar to the popular BitTorrent protocol [10], which makes use of a full mesh with a subset of peers being exposed to each peer. This approach has been used in many systems such as CoolStreaming [11], PPLive [4], QQLive [5] and TVAnts [12]. A more recent modification is to use random linear network coding techniques [13] to make block selection simpler [14], [15], [16] in the wired and wireless context.

In this paper, our main objective is to characterize the amount of buffering needed for a target probability of frame freezing over the duration of the video. We consider a simple model in which network coding is used across the packets of a block. A wireless user can obtain coded packets from multiple sources (other users and servers). However, since the wireless channel is unreliable, packets cannot be obtained deterministically. Thus, our question is *how much should we buffer prior to playout in order to account for wireless channel variations?*

There is significant work in the space of P2P content distribution, and we discuss a subset of this work below. Lower-bounds and achievable limits on the delay experienced in P2P file distribution are considered in [17], [18], [19], [20], [21], [22]. The objective here is to quantify the time needed for all users interested in a file to obtain it. In the streaming context, [23] develops upper bounds on performance using meshes and trees, while [9] contains simulation studies. Closer to our work, [24], [25], [26], [27] develop analytical models on the tradeoff between the steady state probability of missing a block, and buffer size under different block selection policies for live streaming in a full mesh P2P network with deterministic channels. In comparison to these pieces of work, we focus on a very different scenario of streaming of pre-prepared content over unreliable wireless channels. Further, our whole analysis is on transient effects—we are interested in the first time that a frame freezes as a function of the initial amount of buffering.

II. System Overview

Consider a single user playing a media file in a streaming manner. Generally, media files are divided into blocks of frames, and the media player applications are such that they require a complete block to be able to play any of the frames of the block. At the application layer, the user requests blocks from the server (or other peers). The application layer at the server feeds the requested block to the network layer at which the block is divided to multiple packets and sent to the user. Figure 1 illustrates this process.

In a P2P streaming system, the packets in each block can be received from different peers. Traditional mesh-based pull

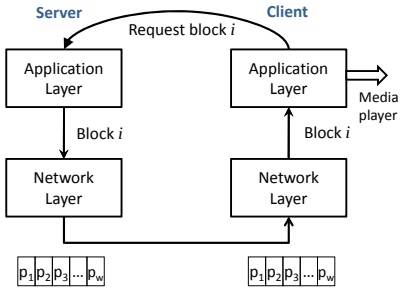

Fig. 1. The media player (application layer) requires complete blocks. At the network layer each block is divided into packets and delivered.

streaming strategies involve each peer storing a bitmap of the available packets in each block, and requesting the missing packets. This approach can result in receiving duplicate packets from different peers or wasting resources communicating with those without any packets useful to the receiver. In the following, we intuitively demonstrate how to use network coding to alleviate these problems.

A. Network Coding for Streaming

Instead of requesting individual packets of each block i from different servers (peers), the receiver only requests a *degree of freedom* of block i from different servers. In this scheme, when a server is requested a degree of freedom of block i, it forms a coded packet as a linear combination of all the packets it has in block i. The coefficients of each linear combination are chosen uniformly at random from a Galois field of size q. The coded packets delivered to the receiver can be thought of linear equations, where the unknowns are the original packets in block i. The original packets in block i can be recovered by solving a system of linear equations if it is full rank. It can be shown that if the field size q is large enough, the received linear equations are linearly independent with very high probability [13]. Therefore, for recovering a block of W packets, it is sufficient to receive W coded packets from different servers.

Using network coding eliminates the need for keeping track of the exact map of the available packets in each block and asking only for the missing packets. Moreover, since each received coded packet is linearly independent from the previous ones with high probability, it is very unlikely to receive duplicate (redundant) packets. Further, in a system with limited storage at the servers, it may not be possible to store all packets corresponding to each block. However, by storing random linear combination of the packets at the servers, we can deliver a new degree of freedom to the receiver upon request (see [**?**] for several other practical benefits of network coding for P2P streaming).

Finally, note that network coding does not introduce additional decoding delay for each block of the media file. This is so since the media player application cannot play an uncomplete block, and for each block of W packets, W degrees of freedom are required. These degrees of freedom can be of the form of coded or distinct uncoded packets.

Next, we describe an abstraction of a streaming system with network coding, which allows us to present a simple dynamics for the receiver's buffer.

III. System Model and QoE Metrics

Consider a single user receiving a media file from various peers it is connected to. Each peer could be a wireless access point or another wireless user operating as a server. We assume that the video file consists of T packets that are divided into blocks of length W. Each server sends random linear combinations of the packets within the current block to the receiver. As we discussed in the preceding section, if the linear combination coefficients are selected from a Galois field of size q, for large enough q no redundant packet will be delivered to the receiver with very high probability. Here we assume the block size W is small compared to the total length of the file, but large enough to ignore the boundary effects of moving from one block to the next. Assume that time is continuous and the arrival process from each peer is an Poisson process independent of other arrival processes. Since no redundant packet is delivered from different peers, we can combine the arrival processes into one Poisson process of rate R, where we normalize the play rate to one, i.e., it takes one unit of time to play a single packet. Therefore, the system boils down to a single server-single receiver system. It is worth mentioning that we owe the simplicity of this model to network coding that eliminates the need for coordination and packet reordering. We assume that the parameter R is known at the receiver. In our model, the receiver first buffers D packets from the beginning of the file, and then starts the playback.

For every block of W packets, we need to receive W coded packets from the server. Each block cannot be decoded and played until all W packets of that block have arrived. Therefore, having received coded packets does not immediately yield interruption-free playback. However, we can treat the coded packets as if they are immediately decodable by the following argument. Assume that whenever a block of W coded packets is decoded, the decoded packets replace the coded ones in the buffer. If at any instance there are at least W packets in the buffer, then there is at least one decoded packet in the buffer. This is so since either the first W packets in the buffer belong to the same block, or they belong to two different blocks. In the former case, the packets of the block can be decoded, and in the latter case, the first block of the two must be already decoded; otherwise, the next block would not be sent from the server. Therefore, the dynamics of the receiver's buffer can be described as follows

$$Q(t) = \max\{D + A(t) - t, 0\}, \tag{1}$$

where D is the initial buffer size and $A(t)$ is a Poisson process of rate R. In this work, we ignore the integrality constraint of the buffer size for simplicity of notation. We declare an interruption in playback when the buffer size reaches the threshold W. Again for simplicity of notation, we assume that an *extra* block of W packets is initially buffered (not taken into account in D). Hence, we can declare an interruption in streaming when the buffer size reaches zero before reaching

the end of the file. More precisely, let

$$\begin{aligned}\tau_e &= \inf\{t : Q(t) \le 0\},\\ \tau_f &= \inf\{t : Q(t) \ge T - t\}.\end{aligned} \tag{2}$$

The video streaming is interrupted if and only if $\tau_e < \tau_f$.

In this work we consider the following metrics to quantify Quality of User Experience (QoE) of the video streaming. The first metric is the initial waiting time before the playback starts. This is directly captured by the initial buffer size D. Another metric that affects QoE is the probability of experiencing an interruption during the playback, which is denoted by

$$p(D) = \mathbf{Pr}\{\tau_e < \tau_f\}, \tag{3}$$

where τ_e and τ_f are defined in (2). In our model, user expects to have an interruption-free experience with probability higher than a desired level $1-\epsilon$. Note that there is a fundamental trade-off between the interruption probability and the initial buffer size. For example, owing to the randomness of the arrival process, in order to have zero probability of interruption it is necessary to fully download the file, i.e., $D = T$. Nevertheless, we need to buffer only a small fraction of the file if user tolerates a positive probability of interruption. These trade-offs and their relation to system parameters R and T are addressed in the following section.

IV. Optimal QoE Trade-offs

In this section, we obtain bounds on the optimal trade-off curve of the QoE metrics introduced in the preceding section as a function of the system parameters. In other words, we would like to obtain the smallest initial buffer size so that the interruption probability is below a desired level ϵ, which is denoted by

$$D^*(\epsilon) = \min\{D \ge 0 : p(D) \le \epsilon\}, \tag{4}$$

where $p(D)$ is the interruption probability defined in (3). Note that in general $p(D)$ and hence $D^*(\epsilon)$ depend on the arrival rate R and the file size T which are assumed to to be known and constant. In the following we characterize the optimal trade-off between the initial buffer size and the interruption probability by providing bounds on $D^*(\epsilon)$. An upper bound (achievability) on $D^*(\epsilon)$ is particularly useful, since it provides a sufficient condition for desirable user experience. A lower bound (converse) of $D^*(\epsilon)$ provides a necessary condition on the initial buffer size for a desirable level ϵ of interruption probability. Let us first introduce some useful lemmas.

Lemma 1. *Let $X(t) = e^{-rQ(t)}$, where $Q(t)$ is given by (1). Then for every $r \ge 0$ such that $\gamma(r) = r + R(e^{-r}-1) \ge 0$, $X(t)$ is a sub-martingale with respect to the canonical filtration $\mathcal{F}_t = \sigma(X(s), 0 \le s \le t)$.*

Proof: For every t, $|X(t)| \le 1$. Hence, $X(t)$ is uniformly integrable. It remains to show that for every $t \ge 0$ and $h > 0$,

$$\mathbf{E}[X(t+h)|\mathcal{F}_t] \ge X(t) \quad \text{a.s.} \tag{5}$$

The left-hand side of (5) can be expressed as

$$\begin{aligned}\mathbf{E}[X(t+h)|\mathcal{F}_t] &= \mathbf{E}\Big[e^{-r(Q(t+h)-Q(t))}\Big|\mathcal{F}_t\Big]X(t)\\ &\ge \mathbf{E}\Big[e^{-r(A(t+h)-A(t))}\Big|\mathcal{F}_t\Big]e^{rh}X(t)\\ &\overset{(a)}{=} \mathbf{E}\big[e^{-rA(h)}\big]e^{rh}X(t)\\ &\overset{(b)}{=} e^{h(r+R(e^{-r}-1))}X(t) = e^{h\gamma(r)}X(t),\end{aligned}$$

where (a) follows from independent increment property of the Poisson process, and (b) follows from the fact that $A(t)$ is a Poisson random variable. Now, it is immediate to verify (5) for any r with $\gamma(r) \ge 0$. ∎

Next, we use Doob's maximal inequality to bound the interruption probability.

Lemma 2. *Let $p(D)$ be the interruption probability given the initial buffer size D. Then, for any $r \ge 0$ with $\gamma(r) = r + R(e^{-r}-1) \ge 0$*

$$p(D) \le e^{-rD+T\gamma(r)}, \quad \textit{for all } D, T, R \ge 0. \tag{6}$$

Proof: By definition of $p(D)$ in (3), we have

$$\begin{aligned}p(D) &= \mathbf{Pr}\{\tau_e < \tau_f\}\\ &\le \mathbf{Pr}\{\tau_e \le T\} = \mathbf{Pr}\Big\{\inf_{0\le t\le T} Q(t) \le 0\Big\}\\ &= \mathbf{Pr}\Big\{\sup_{0\le t\le T} e^{-rQ(t)} \ge 1\Big\}\\ &\overset{(a)}{\le} \mathbf{E}[e^{-rQ(T)}] = \mathbf{E}[e^{-r(D+A(T)-T)}]\\ &= e^{-r(D-T)}e^{RT(e^{-r}-1)} = e^{-rD+T\gamma(r)},\end{aligned}$$

where (a) holds by applying Doob's maximal inequality to the non-negative sub-martingale $X(t) = e^{-rQ(t)}$. Note that $X(t)$ is a sub-martingale for all r with $\gamma(r) \ge 0$ by Lemma 5. ∎

Lemma 3. *Define $\bar{r}(R)$ as the largest root of $\gamma(r) = r + R(e^{-r}-1)$, i.e.,*

$$\bar{r}(R) = \sup\{r : \gamma(r) = 0\}. \tag{7}$$

We have

$$\bar{r}(R) = 0, \qquad \textit{if } 0 \le R \le 1, \tag{8}$$

$$\frac{2(R-1)}{R} \le \bar{r}(R) \le 2(R-1), \qquad \textit{if } 1 \le R \le 2, \tag{9}$$

$$R - 1 \le \bar{r}(R) \le R \le 2(R-1), \qquad \textit{if } R \ge 2. \tag{10}$$

Proof: See Appendix. ∎

Next, we provide sufficient conditions on the initial buffer size to avoid interruptions with high probability for different regimes of the arrival rate.

Theorem 1. [Achievability] *Let $D^*(\epsilon)$ be defined as in (4), and $\bar{r}(R)$ be the largest root of $\gamma(r)$ defined in (7). Then*

(a) *For all $R > 1$,*

$$D^*(\epsilon) \le \frac{1}{\bar{r}(R)} \log\big(\frac{1}{\epsilon}\big). \tag{11}$$

(b) *For all* $0 \le R \le 1 + \left(\frac{1}{2T}\log\left(\frac{1}{\epsilon}\right)\right)^{\frac{1}{2}}$,

$$D^*(\epsilon) \le \min\Big\{\frac{1}{\bar{r}(R)}\log\big(\frac{1}{\epsilon}\big), \; T(1-R) + \Big(2TR\log\big(\frac{1}{\epsilon}\big)\Big)^{\frac{1}{2}}\Big\}. \quad (12)$$

Proof: First, note that for any upper bound $\bar{p}(D)$ of the interruption probability $p(D)$, any feasible solution of the problem

$$\bar{D}(\epsilon) = \min\{D \ge 0 : \bar{p}(D) \le \epsilon\} \quad (13)$$

provides an upper bound on $D^*(\epsilon)$. This is so since the optimal solution of the above problem is feasible in the minimization problem (4). If the problem in (13) is infeasible, we use the convention $\bar{D}(\epsilon) = \infty$, which is a trivial bound on $D^*(\epsilon)$. The rest of the proof involves finding the tightest bounds on $p(D)$ and solving (13).

Part (a): By Lemma 2, for $r = \bar{r}(R)$, we can write

$$p(D) \le \bar{p}_a(D) = e^{-\bar{r}(R)D}, \quad \text{for all } D, T, R \ge 0.$$

Solving $\bar{p}_a(D) = \epsilon$ for D gives the result of part (a). Since $\bar{r}(R) = 0$ for $R \le 1$ (cf. Lemma 3), this bound is not useful in that range.

Part (b): First, we claim that for all $D \ge T(1 - R + \bar{r}(R))$,

$$p(D) \le \bar{p}_b(D) = e^{-\frac{1}{2}TRz^2},$$

where $z = 1 - \frac{1}{R}\big(1 - \frac{D}{T}\big)$. We use Lemma 2 with $r = r^* = -\log\big(\frac{1}{R}\big(1 - \frac{D}{T}\big)\big)$ to prove the claim. Note that $r^* \ge 0$, because $D \ge T(1 - R)$. In order to verify the second hypothesis of Lemma 2, consider the following

$$\begin{aligned} R(e^{-r^*} - e^{-\bar{r}(R)}) &= \bar{r}(R) + R(e^{-r^*} - 1) - \gamma(\bar{r}(R)) \\ &= \bar{r}(R) - R + (1 - \frac{D}{T}) \\ &= \frac{1}{T}\Big[T(1 - R + \bar{r}(R)) - D\Big] \le 0, \end{aligned}$$

where the inequality follows from the hypothesis of the claim. Thus, $r^* \ge \bar{r}(R)$, and by definition of $\bar{r}(R)$ in (7), we conclude that $\gamma(r^*) \ge 0$. Now, we can apply Lemma 2 to get

$$\begin{aligned} p(D) &\le e^{-r^*D + T\gamma(r^*)} \\ &\overset{(a)}{=} e^{TR\left(\frac{1}{R}(1-\frac{D}{T})r^* - (1-e^{-r^*})\right)} \\ &\overset{(b)}{=} e^{TR\left(-(1-z)\log(1-z) - z\right)} \\ &\overset{(c)}{\le} e^{-\frac{1}{2}TRz^2}, \end{aligned}$$

where (a) and (b) follow from the definition of $\gamma(r)$ and z. Further, (c) holds by Lemma 4 of the Appendix. Therefore, the claim holds.

Now, let $\bar{D} = T(1 - R) + \Big(2TR\log\big(\frac{1}{\epsilon}\big)\Big)^{\frac{1}{2}}$. It is straightforward to check that $p(\bar{D}) \le \bar{p}_b(\bar{D}) = \epsilon$, if $\bar{D} \ge T(1 - R + \bar{r}(R))$. This result follows from Lemma 2 and noting that for $R \le 1$, $\bar{r}(R) = 0$ (cf. Lemma 3). Then for all $1 \le R \le 1 + \left(\frac{1}{2T}\log\left(\frac{1}{\epsilon}\right)\right)^{\frac{1}{2}}$, we have

$$\begin{aligned} \bar{D} - T(1-R) &= \Big(2TR\log\big(\frac{1}{\epsilon}\big)\Big)^{\frac{1}{2}} \\ &\ge 2T\Big(\frac{1}{2T}\log\big(\frac{1}{\epsilon}\big)\Big)^{\frac{1}{2}} \\ &\overset{(d)}{\ge} 2T(R-1) \overset{(e)}{\ge} T\bar{r}(R), \end{aligned}$$

where inequality (d) follows from the hypothesis of Part (b), and inequality (e) is true by Lemma 3. Therefore, $D^*(\epsilon) \le \bar{D}$ for all $R \le 1 + \left(\frac{1}{2T}\log\left(\frac{1}{\epsilon}\right)\right)^{\frac{1}{2}}$. Note that, the upper bound that we obtained in Part (a) is also valid for all R. Hence, the minimum of the two gives the tightest bound. ■

When the arrival rate R is smaller than the playback rate, the upper bound in Theorem 1 consists of two components. The first term, $T(1 - R)$, compensates the expected number of packets that are required by the end of $[0, T]$ period. The second component, $\Big(2TR\log\big(\frac{1}{\epsilon}\big)\Big)^{\frac{1}{2}}$, compensates the randomness of the arrivals to avoid interruptions with high probability. Note that this term increases by decreasing the maximum allowed interruption probability, and it would be zero for a deterministic arrival process. For the case when the arrival rate is larger than the playback rate, the minimum required buffer size does not grow with the file size. By continuity of the probability measure, we can show that the upper bound in Theorem 1 remains bounded for infinite file sizes. This is so since the buffer size in (1) has a positive drift. Hence, if there is no interruption at the beginning of the playback period, it becomes more unlikely to happen later.

In the following, we show that the upper bounds presented in Theorem 1 are *asymptotically tight*, by providing lower bounds on the minimum required buffer size $D^*(\epsilon)$, for different regimes of the arrival rate R. Let us first define the notion of a tight bound.

Definition 1. Let $\hat{D}$ be a lower or upper bound of the minimum buffer size $D^*(\epsilon)$ that depends on the file size T. The bound $\hat{D}$ is an *asymptotically tight* bound if $\frac{|\hat{D} - D^*(\epsilon)|}{D^*(\epsilon)}$ vanishes as T goes to infinity.

Theorem 2. [Converse] *Let $D^*(\epsilon)$ be defined as in (4), and $\bar{r}(R)$ be the largest root of $\gamma(r)$ defined in (7). Then*

(a) *For all $R > 1$,*

$$D^*(\epsilon) \ge -\frac{1}{\bar{r}(R)}\log\Big(\epsilon + 2e^{-\frac{(R-1)^2}{2(R+1)}T}\Big). \quad (14)$$

(b) *For each $0 \le R \le 1$ and $\epsilon \le \frac{1}{16}$, if $T \ge C\log\left(\frac{1}{\epsilon}\right)$ then*

$$D^*(\epsilon) \ge T(1-R) + \frac{1}{2}\Big(2TR\log\big(\frac{1}{\epsilon}\big)\Big)^{\frac{1}{2}}, \quad (15)$$

where C is a constant that only depends on R.

Proof: We do not present the proof due to space limitation. See [**?**] for a complete proof. ■

Note that the assumption $\epsilon \le \frac{1}{16}$ in part (b) of Theorem 2 is necessary for the result to hold; otherwise, we can show that

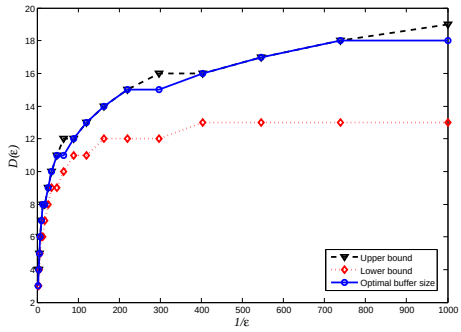

Fig. 2. The minimum buffer size $D^*(\epsilon)$ as a function of the interruption probability.

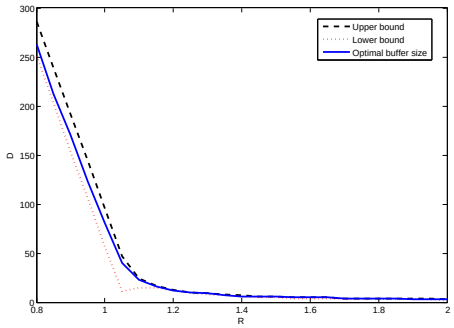

Fig. 3. The minimum buffer size $D^*(\epsilon)$ as a function of the arrival rate R.

$D^*(\epsilon) < T(1-R)$ for a large interruption probability ϵ. In the limit $\epsilon = 1$, it is clear that $D^*(\epsilon) = 0$. Nevertheless, since we are interested in *avoiding* interruptions, we do not study this regime of the interruption probabilities. Comparing the lower bounds obtained in Theorem 2 with the upper bounds obtained in Theorem 1, we observe that they demonstrate a similar behavior as the system parameters T and R change. Now, we can show that the obtained bounds are asymptotically tight.

Corollary 1. *The upper bounds and lower bounds of $D^*(\epsilon)$ given by Theorems 1 and 2 are asymptotically tight, if $R > 1$, or $R < 1$ and $\epsilon \leq \frac{1}{16}$.*

Proof: Let D_l and D_u be lower and upper bounds of $D^*(\epsilon)$, respectively. By Definition 1, for D_l or D_u to be asymptotically tight, it is sufficient to show $\frac{D_u - D_l}{D_l}$ goes to zero as T grows. It is straightforward to verify this claim, using the upper and lower bounds presented in Theorem 1 and Theorem 2, and taking the limit as T goes to infinity. ■

Next, we numerically obtain the optimal trade-off curve between the interruption probability and initial buffer size, and compare the results with the bounds derived earlier.

V. Numerical Results

We use MATLAB simulations to compute the minimum initial buffer size $D^*(\epsilon)$ for a given interruption probability ϵ in various scenarios. Towards this goal, we start from a small initial buffer size D, and for each D we compute the interruption probability $p(D)$ via Monte-Carlo method. We increase D until the constraint $p(D) \leq \epsilon$ is satisfied. Since $p(D)$ is monotonically decreasing in D, this gives the minimum required buffer size. Here, we restrict D to take only integer values, and round each upper bound value up to the nearest integer, and each lower bound value down to the nearest integer.

Figure 2 shows the minimum required buffer size $D^*(\epsilon)$ as well as the upper and lower bounds given by Theorems 1 and 2 as a function of $\frac{1}{\epsilon}$, where the arrival rate is fixed to $R = 1.2$ and the file size $T = 500$. We observe that the numerically computed trade-off curve closely matches our analytical results.

Figure 3 plots the minimum required buffer size $D^*(\epsilon)$ as well as the upper and lower bounds given by Theorems 1 and 2 versus the arrival rate R, where $\epsilon = 10^{-2}$ and the file size is fixed to $T = 10^3$. Note that when the arrival rate is almost equal or less than the play rate, increasing the arrival rate can significantly improve the initial buffering delay. However, for larger arrival rates $D^*(\epsilon)$ is small enough such that increasing R does not help anymore. This could provide a guidance for resource allocation among multiple users with certain QoE requirements.

VI. Conclusions

We studied the problem of media streaming with focus on Quality of user Experience (QoE) metrics and trade-offs. The QoE metrics that we considered in this work are the probability of interruption in media playback and initial waiting time before starting the playback.

In our system, the user can receive parts of the media file from multiple sources by requesting packets in each block of the file. We demonstrated that sending a random linear

combinations of the packets within each block of the media file simplifies the packet selection strategies of the P2P systems, and solves the duplicate packet reception issue. Moreover, it allowed us to describe the receiver's buffer dynamics as an M/D/1 queue, and characterize the trade-off between the QoE metrics for different ranges of the system parameters. We presented tight upper and lower bounds on the minimum initial buffering required to achieve a desired level of interruption probability. Finally, our numerical results confirmed that the optimal trade-off curve demonstrate a similar behavior to the one predicted by our bounds.

This work is the first step in analytical characterization of QoE trade-offs in media streaming applications. It is essential to take into account each user's preferences on the interruption probability and initial waiting time, when performing resource allocation among multiple users. An interesting extension to this work would be to obtain optimal resource allocation policies to satisfy certain user preferences. We shall study this problem in future works.

APPENDIX

Proof of Lemma 3: *Case I* $(0 \leq R \leq 1)$: First note that $\gamma(r)$ is a continuously differentiable function, and $\gamma(0) = 0$. For each $R < 1$, we have $\gamma'(r) > 0$ for all $r \geq 0$. Therefore, $\gamma(r) > 0$ for all $r > 0$, i.e., $\bar{r}(R) = 0$ for each $R < 1$.

Case II $(1 \leq R \leq 2)$: By definition of $\bar{r}(R)$ in (7),

$$\begin{aligned} 0 = \gamma(\bar{r}(R)) &= \bar{r}(R) + R(e^{-\bar{r}(R)} - 1) \\ &\leq \bar{r}(R) + R(-\bar{r}(R) + \frac{\bar{r}^2(R)}{2}). \end{aligned}$$

Rearranging the terms in the above relation, gives the lower bound in (9). We show the upper bound in two steps. First, we show that $\gamma(2(R-1)) > 0$ for $R > 1$, then we verify that $\gamma(r) \geq 0$ for all $r \geq 2(R-1)$. These two facts imply that $\gamma(r) > 0$ for all $r \geq 2(R-1)$, i.e., $\bar{r}(R) \leq 2(R-1)$. The first step can be verified by noting that

$$\gamma(2(R-1))\big|_{R=1} = 0, \quad \frac{\partial}{\partial R}\gamma(2(R-1)) > 0.$$

It is also straightforward to show that

$$\frac{\partial}{\partial r}\gamma(r) > 0, \quad \text{for all } r \geq \log(R), \tag{16}$$

which immediately yields the second step by noting $r \geq 2(R-1) \geq \log(R)$.

Case III $(R \geq 2)$: We use a similar technique as in the preceding case. The upper bound is immediate by the following facts:

$$\gamma(R) = Re^{-R} > 0, \quad \frac{\partial}{\partial r}\gamma(r) > 0, \quad \text{for all } r \geq R.$$

It is also straightforward to check that $\gamma(R-1) < 0$ for all $R \geq 2$. Moreover, note that $\gamma(R) > 0$. Therefore, by the mean value theorem, $\gamma(r)$ has a root in $[R-1, R]$, i.e., $\bar{r}(R) \geq R-1$.

Lemma 4. *For all* $0 \leq z < 1$*, the following relation holds:*

$$-(1-z)\log(1-z) - z \leq -\frac{z^2}{2}. \tag{17}$$

Proof: Let $f(z) = -(1-z)\log(1-z) - z + \frac{z^2}{2}$. $f(z)$ is a continuously differentiable function on $[0, 1)$. Moreover, $f(0) = 0$, and $f'(z) = \log(1-z) + z \leq 0$. Therefore, $f(z) \leq f(0) = 0$, for all $z \in [0, 1)$. ■

REFERENCES

[1] C. Fraleigh, S. Moon, B. Lyle, C. Cotton, M. Khan, D. Moll, R. Rockell, T. Seely, and C. Diot, "Packet-level traffic measurements from the Sprint IP backbone," *IEEE Network Magazine*, vol. 17, no. 6, pp. 6–16, 2003.

[2] K. P. Gummadi, R. J. Dunn, S. Saroiu, S. D. Gribble, H. M. Levy, and J. Zahorjan, "Measurement, modeling, and analysis of a peer-to-peer file-sharing workload," in *Proc. SOSP*, Oct. 2003.

[3] C. Labovitz, D. McPherson, and S. Iekel-Johnson, "2009 Internet Observatory report," in *NANOG-47*, October 2009.

[4] "PPLive," http://www.pplive.com/, 2009.

[5] "QQLive," http://www.qqlive.com/, 2009.

[6] V. N. Padmanabhan, H. J. Wang, P. A. Chou, and K. Sripanidkulchai, "Distributing Streaming Media Content Using Cooperative Networking," in *Proceedings of The 12th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV '02)*, Miami, FL, May 2002.

[7] E. Setton and J. Apostolopoulos, "Towards Quality of Service for Peer-to-Peer Video Multicast," in *Proc. of IEEE International Conference on Image Processing (ICIP)*, San Antonio, TX, September 2007.

[8] S. Liu, R. Zhang-Shen, W. Jiang, J. Rexford, and M. Chiang, "Performance bounds for peer-assisted live streaming," in *Proc. ACM SIGMETRICS*, June 2008.

[9] N. Magharei, R. Rejaie, and Y. Guo, "Mesh or multiple-tree: A comparative study of live p2p streaming approaches," in *Proc. IEEE INFOCOM*, Anchorage, AK, May 2007.

[10] B. Cohen, "Incentives to build robustness in BitTorrent," in *Workshop on Economics of Peer-to-Peer Systems*, Berkeley, CA, June 2003.

[11] X. Zhang, J. Liu, B. Li, and T.-S. P. Yum, "Coolstreaming/donet: A data-driven overlay network for efficient live media streaming," in *Proc. IEEE INFOCOM*, Miami, FL, March 2005.

[12] "TVAnts," http://www.tvants.com/, 2009.

[13] T. Ho, R. Koetter, M. Médard, M. Effros, J. Shi, and D. Karger, "A random linear network coding approach to multicast," *IEEE Transactions on Information Theory*, vol. 52, pp. 4413–4430, 2006.

[14] C. Feng and B. Li, "On large-scale peer-to-peer streaming systems with network coding," in *Proceedings of the 16th ACM international conference on Multimedia*, Vancouver, Canada, October 2008.

[15] D. Lucani, F. Fitzek, M. Médard, and M. Stojanovic, "Network coding for data dissemination: It is not what you know but what your neighbors don't know," in *Proc. RAWNET*, Seoul, Korea, June 2009.

[16] M. Wang and B. Li, "R2: Random push with random network coding in live peer-to-peer streaming," *IEEE JSAC, Special Issue on Advances in Peer-to-Peer Streaming Systems*, vol. 25, pp. 1655–1666, 2007.

[17] D. Qiu and R. Srikant, "Modeling and performance analysis of BitTorrent-like peer-to-peer networks," in *Proceedings of the ACM SIGCOMM*, Portland, Oregon, USA, August 2004.

[18] S. Deb, M. Médard, and C. Choute, "Algebraic gossip: a network coding approach to optimal multiple rumor mongering," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2486–2507, June 2006.

[19] S. Sanghavi, B. Hajek, and L. Massoulie, "Gossiping with multiple messages," in *Proc. IEEE INFOCOM*, Anchorage, AK, May 2007.

[20] M. Vojnovic and L. Massoulie, "Coupon replication systems," *IEEE/ACM Transactions on Networking*, vol. 16, no. 3, pp. 603–616, June 2008.

[21] S. Shakkottai and R. Johari, "Demand Aware Content Distribution on the Internet," *IEEE/ACM Transactions on Networking*, 2009, to appear.

[22] M. Chen, M. Ponec, S. Sengupta, J. Li, and P. A. Chou, "Utility maximization in peer-to-peer Systems," in *Proc. ACM SIGMETRICS*, June 2008.

[23] R. Kumar, Y. Liu, and K. Ross, "Stochastic fluid theory for P2P streaming systems," in *Proc. IEEE INFOCOM*, Anchorage, AK, May 2007.

[24] Y. P. Zhou, D. M. Chiu, and J. C. S. Lui, "A simple model for analyzing P2P streaming protocols," in *Proceedings of IEEE ICNP 2007*, Beijing, China, October 2007.

[25] T. Bonald, L. Massoulié, F. Mathieu, D. Perino, and A. Twigg, "Epidemic live streaming: optimal performance trade-offs," *SIGMETRICS Perform. Eval. Rev.*, vol. 36, no. 1, pp. 325–336, 2008.

[26] B. Q. Zhao, J. C. Lui, and D.-M. Chiu, "Exploring the optimal chunk selection policy for data-driven P2P streaming systems," in *The 9th International Conference on Peer-to-Peer Computing*, 2009.

[27] L. Ying, R. Srikant, and S. Shakkottai, "The Asymptotic Behavior of Minimum Buffer Size Requirements in Large P2P Streaming Networks," in *Proc. of the Information Theory and Applications Workshop (to appear)*, San Diego, CA, February 2010.